Runlinc Intermediate Project 4: Fluid Level Sensor (E32W Version)

Contents

Introduction

Problem

Water is a very precious resource and we want a way to keep track on the water level in tanks, troughs or river levels. Not keeping track of water can lead to water mismanagement which can have dire consequences. Such as flooding and drought. But for now, let's help create a simple water level sensor system for farmers to help farmers know their water tank's water level.

Background

For this project, we will use a characteristic of water which is that water can conduct electricity. Generally, if two conductors are not connected and have a gap between them the circuit of the conductors has no current. Therefore, knowing that water can conduct electricity, we can connect the two conductors within the water. So, using the principle that if the circuit is closed, conductors connected, then we know that at that level where we place the conductor there still have water.

However, because electricity can ignite or cause reaction with some fluids, and additionally, many liquids do not conduct electricity like water; therefore, using fluids as a conductor is generally not implemented, but the principle behind this idea is used.

Ideas

What do we have that can be used to indicate the level of a tank? Let's say the tank only has three levels; full, low or empty. How could we distinguish between these levels? What do we have that could be used to indicate for the farmer what level the tank is currently at?

Plan

Main Setup

To represent a tank, use any container or a cup that can be easily modified, but we highly recommend the use of a cardboard cup. For this project, we will assume you will use a cardboard cup.

On the cup, mark a line about halfway up the cup. Poke a hole (as small as possible) to the marked area then thread a short length of wire, with some length of coating removed, through it (the length of the threaded wire through the hole should be less than the radius of the marked area). Plug any open spaces with play dough, blue tack, or tape. Do the same for another wire but below the marked area, ideally the bottom of the cup. The marked wire is the receiver, while the wire below the marked area is the transmitter.

With this setup, which can be visualized in Figure 1, we can slowly pour water into the cup. When the water is above or on the mark, the Green LED will be turned on while the Red LED is off to indicate that the tank has a fluid level above the mark. If the fluid is below the mark, the Green LED will be turned off while the Red LED will be turned on to indicate that the tank has a fluid level below the mark.

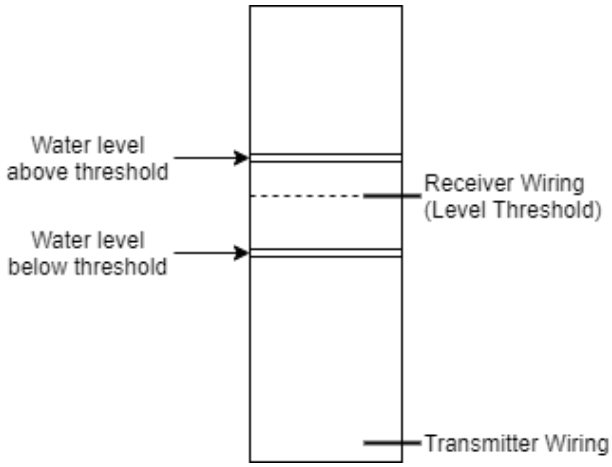


Figure 1: Visualisation of Main Setup

Extension

For the extension, we will need a new cup. This time we will need 4 receivers. We will implement the following water level indicators using its respective margin range for this project as followed in the table below:

Level Indicator	% Margin Range	Condition	LED
Full	95%~100%	Above or Equal Top Receiver	Red
Nearing Full	80%~95%	Above or Equal Lower Top Receiver	Purple
Adequate	20%~80%	Above or Equal Upper Bottom Receiver	Green
Nearing Empty	5%~20%	Above or Equal Bottom Receiver	Yellow
Empty	0%~5%	Below Bottom Receiver	Blue

Use the same method in the previous setup to install the wires to the cup. The level to install the wire from the base to top is roughly 5% (bottom receiver), 20% (upper bottom receiver), 80% (lower top receiver), 95% (top receiver). For example, if your cup is 5cm tall, then install the bottom receiver at roughly 0.25cm away from the base, then install the upper bottom receiver at roughly 1cm away from the base, and same applies to the top receivers. Then install the transmitter below the bottom receiver. You can visualize the setup in the following Figure.

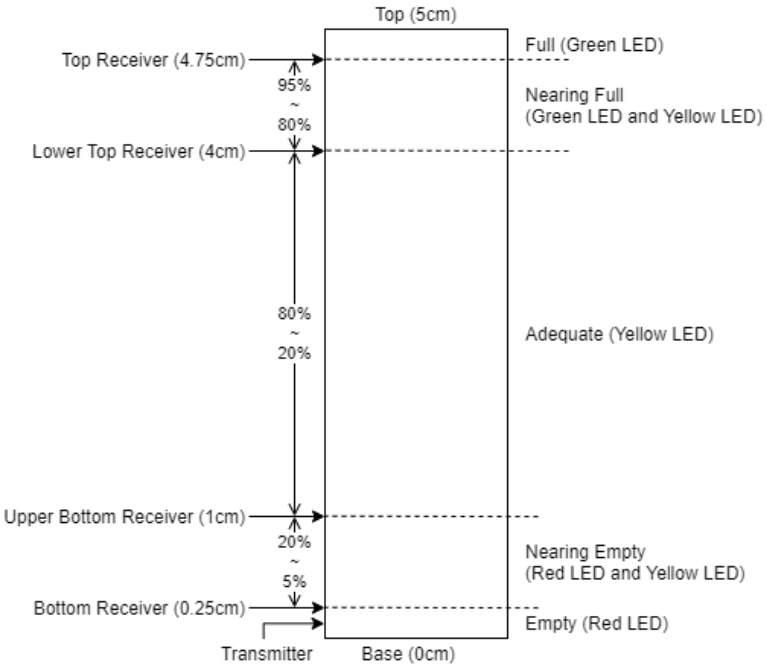


Figure 2: Visualisation of Extension Setup for 5cm cup

runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done inside the browsers compare to programming inside a chip. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

Part A1: Design the Circuit on runlinc (Main Setup)

Note: Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Use the left side of the runlinc web page to construct an input/output (I/O).

For port D18 name it Red and set it as DIGITAL_OUT.
For port D19 name it Green and set it as DIGITAL_OUT.
For port D21 name it Blue and set it as DIGITAL_OUT.
For port D32 name it Receiver and set it as ANALOG_IN.
For port D33 name it Transmitter and set it as DIGITAL_OUT.

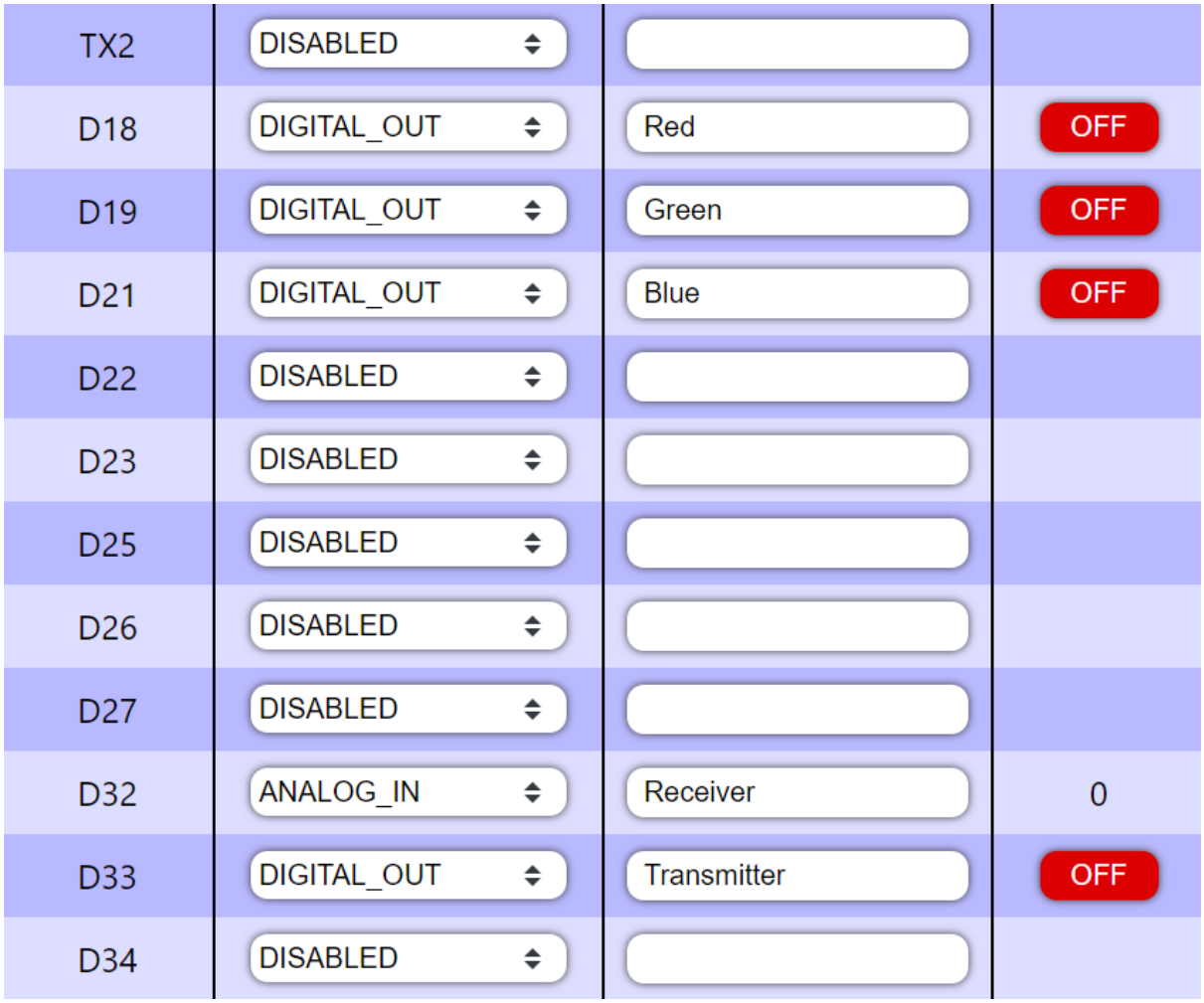


Figure 3: I/O configurations connections

Part B1: Build the Circuit (Main Setup)

Use the STEMSEL E32W board to connect the hardware. For this project we are using both the left and right I/O ports, with **negative port (-ve)** on the outer side, **positive port (+ve)** on the middle and **signal port (s)** on the inner side (as shown below).

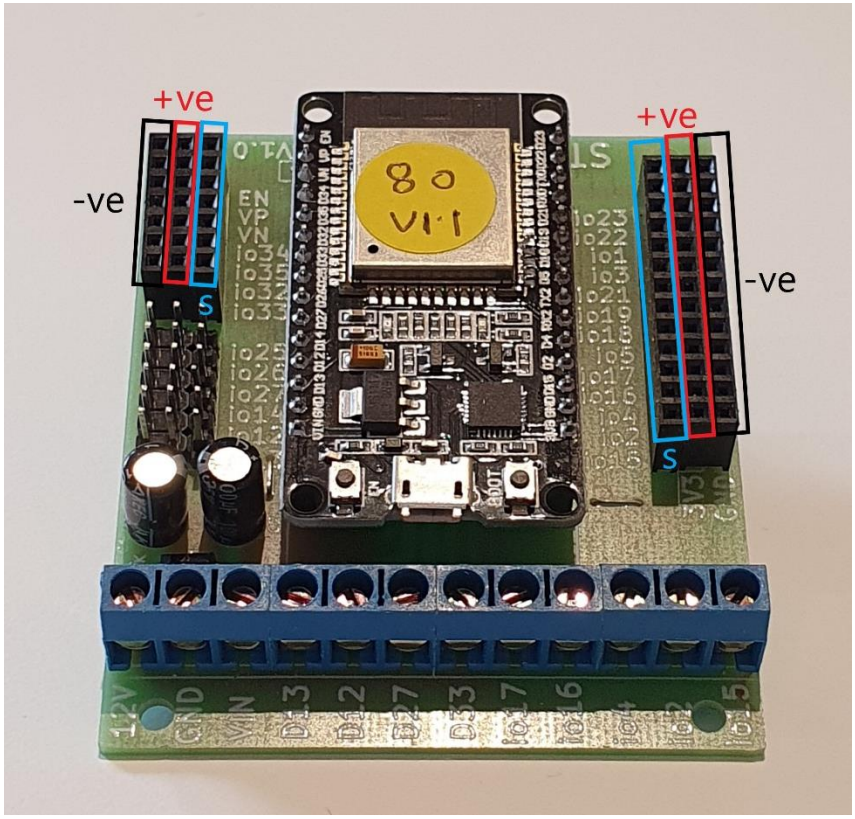


Figure 4: Negative, Positive and Signal port on the E32W board

There is one I/O part and multiple jumper wires we are using for this project, a 4-pin RGB SMD LED module (KY-009). Its respective pins are shown in the figure below.

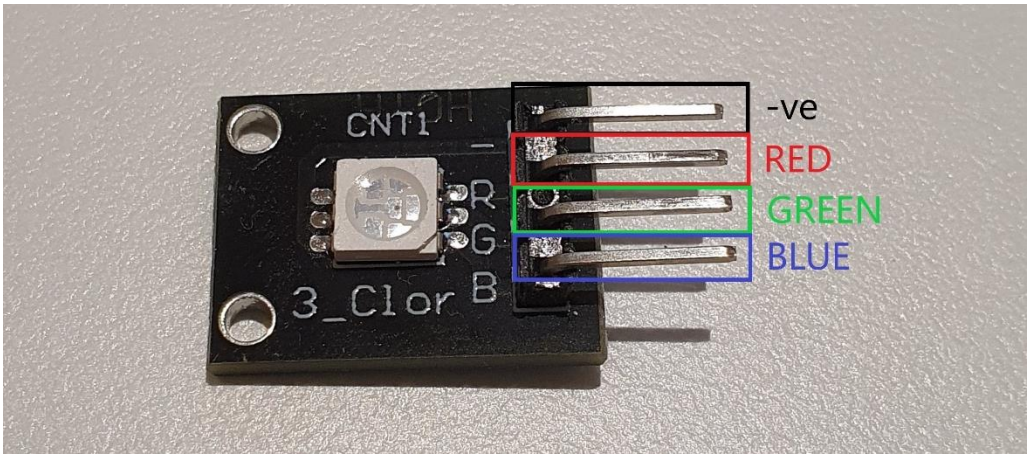


Figure 5: 4-pin RGB SMD LED (KY-009) with its respective pins indicated

Wiring instructions (Main Setup)

To achieve multiple colours, we will be using 4 male-to-female jumper wires to connect the pins from 4-pin RGB SMD LED (KY-009) to our ports, and 3 more male-to-male jumper wires to act as transmitter and receiver.

Note: Male jumper wires have **PIN** end, while female jumper wires **DOES NOT**.

1. Connect 1 female jumper wire to the -ve pin on the KY-009 module, plug the other male end into GND port on io32 (as shown in Figure 6).

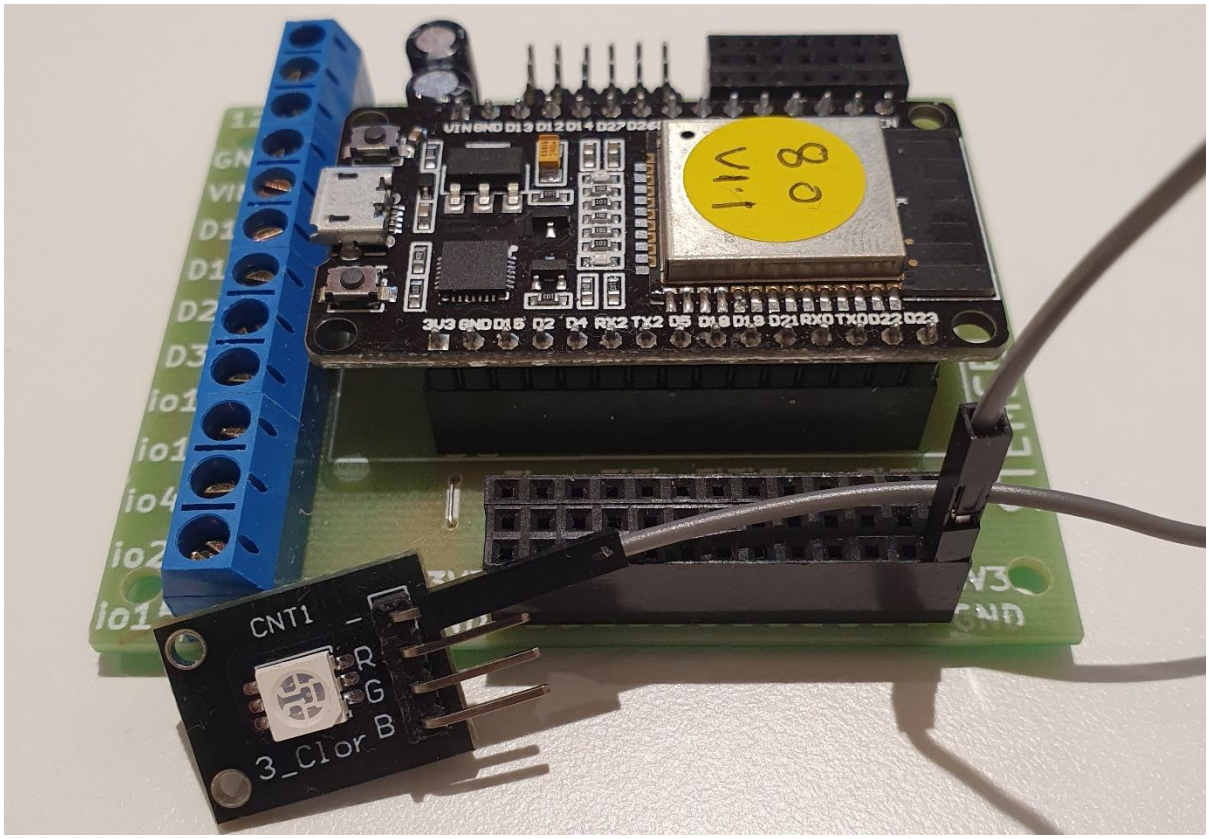


Figure 6: GND connection for KY-009 Module.

2. Connect the 3 other female jumper ends to R, G, B, pins respectively, then plug them into the signal ports of io18(red), io19(green) and io21(blue) (as shown in Figure 7).

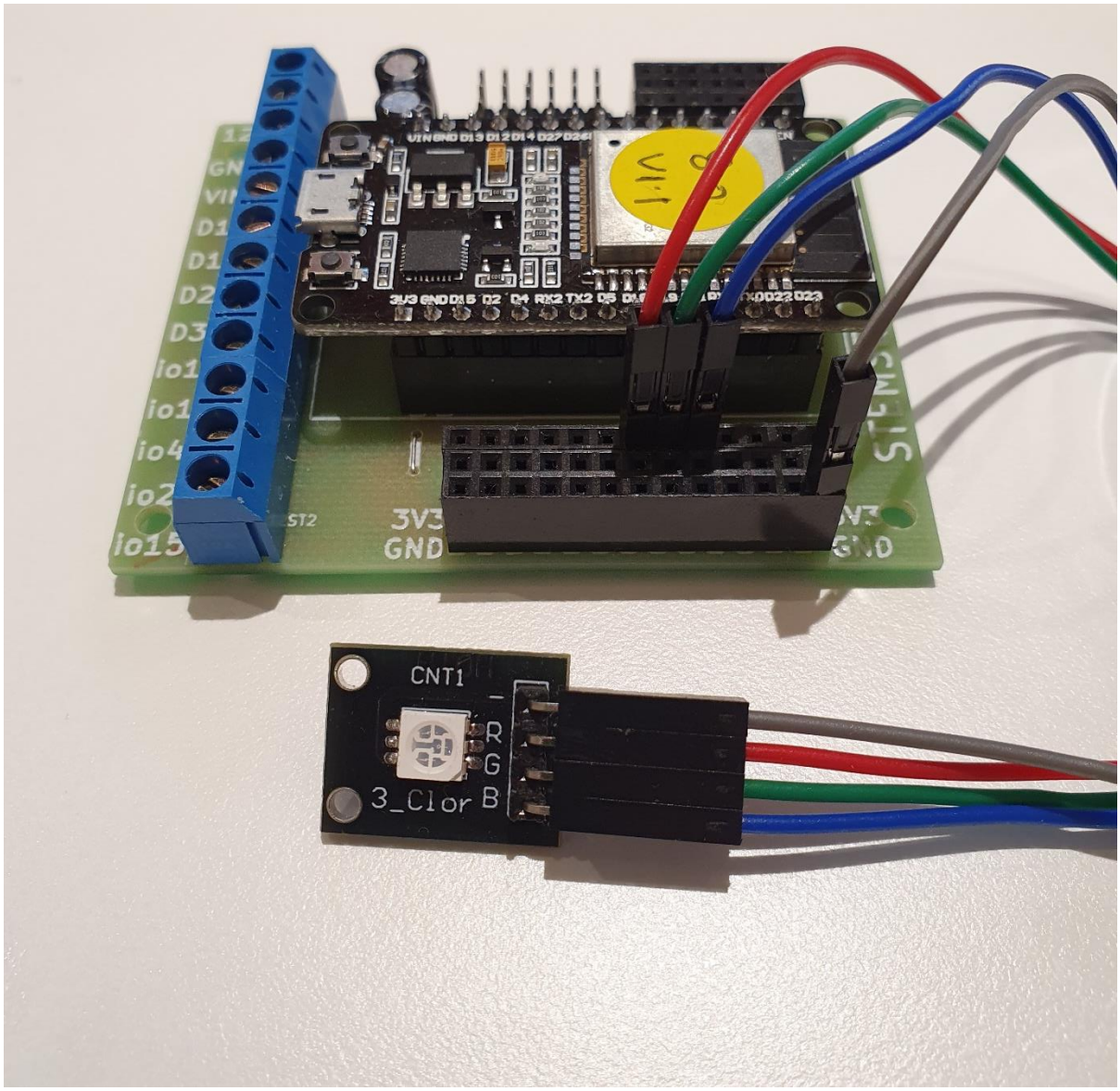


Figure 7: All 4 jumper wires connected to KY-009 module.

3. Take 3 male-to-male jumper wires and plug two of the wires (any male end) into the signal ports io32 and io33, the last wire is connected to GND port on io32 (as shown in Figure 8 & Figure 9).

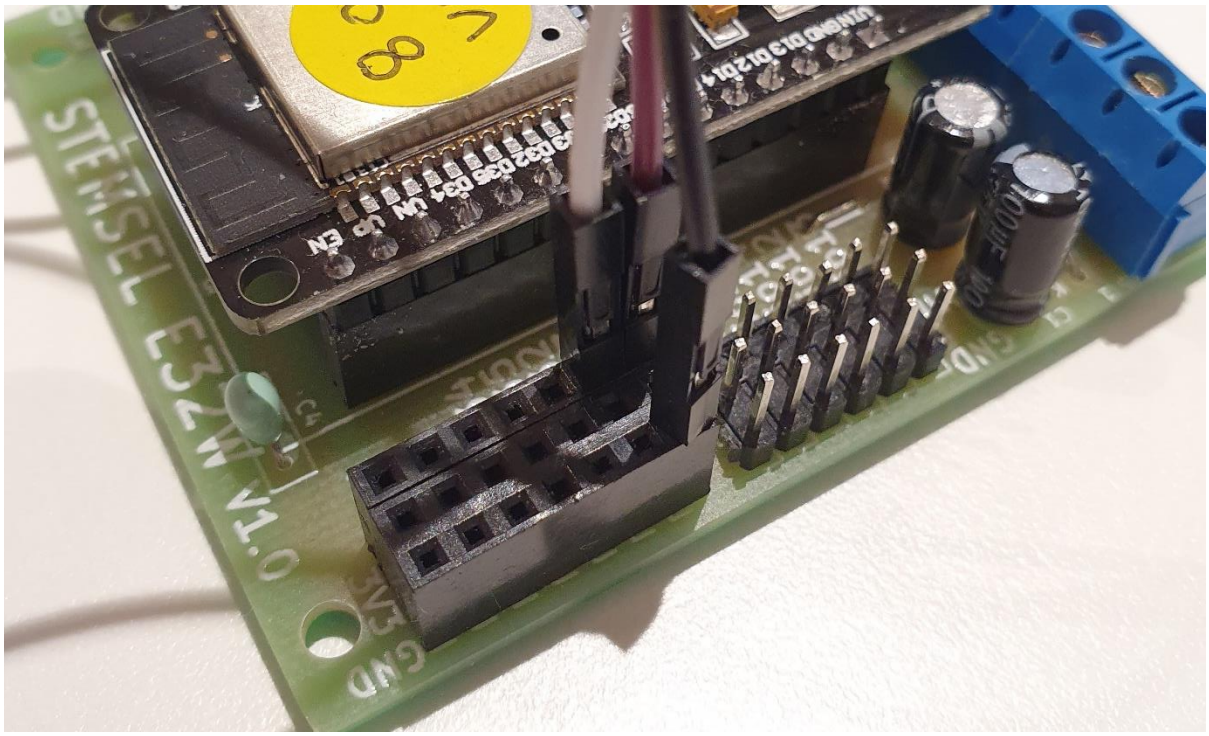


Figure 8: Side view for 3 jumper wires connected to the signal ports of io32 and io33, and GND port of io33.

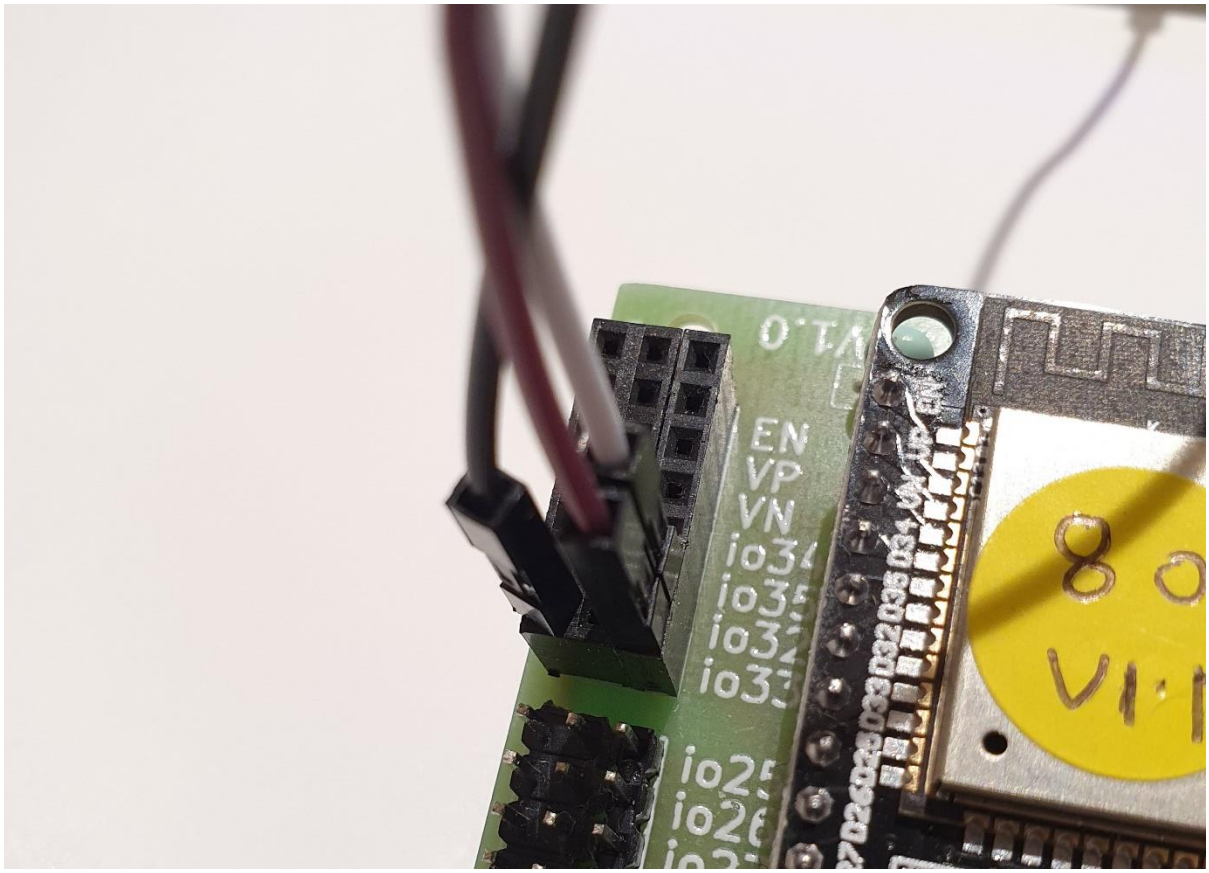


Figure 8: Top view for 3 jumper wires connected to the signal ports of io32 and io33, and GND port of io33.

Part C1: Program the Circuit (Main Setup)

HTML:

We will first set up our HTML page to receive information about the fluid level.

Let's set up the page to have text aligned to the centre with a title: We will add a status text after <h1></h1> that will have their equivalent LED shine.

```
<div style="text-align:center">
 <h1>Fluid Sensor Dashboard by runlinc</h1>
 <br>
 <br>
 Current Status: <font id="Status">Water Level Sensor is loading...</font>
</div>
```

JavaScript:

We will initiate the threshold voltage. This threshold voltage is used to determine if the receiver has formed a closed circuit with the transmitter by comparing its voltage to its minimum threshold voltage. Since runlinc input use 0-255 range for 0-3.3V range then we use 100 for the threshold voltage. And turn on the transmitter.

```
var thresholdVoltage = 100;
turnOn( Transmitter );
```

JavaScript Loop:

Now we will implement the measuring and analysing functions for the sensors. Within the JavaScript Loop block, let's code the receiver's value.

```
receiverValue = analogIn( Receiver );
```

Now we will compare the value to the threshold. If it is bigger or equal to the threshold, then turn on the Green LED and turn off the Red LED and announce the change to the status that the fluid is over the threshold. On the contrary, if it is below the threshold, add some delay to prevent overloading of the chip.

```
if(receiverValue >= thresholdVoltage){
turnOn( Green );
turnOff( Red );
document.getElementById("Status").innerHTML = "The water level is above the marked level.";
}else{
turnOn( Red );
turnOff( Green );
document.getElementById("Status").innerHTML = " The water level is below the marked level.";
}
await mSec(500);
```

Now you can test out the sensor system. You can play around the system by moving the threshold level around by changing the level where you place the receiver.

Note: It is better to place a plate or tray underneath the cup, as leaking could appear and poses a risk for the E32W board.

Expected Result (Main part): No water in the cup.

TX2	DISABLED		
D18	DIGITAL_OUT	Red	ON
D19	DIGITAL_OUT	Green	OFF
D21	DIGITAL_OUT	Blue	OFF
D22	DISABLED		
D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	ANALOG_IN	Receiver	0
D33	DIGITAL_OUT	Transmitter	ON
D34	DISABLED	UpperTopReceiver	
D35	DISABLED	LowerBottomReceiver	
VP	DISABLED	BottomReceiver	
RNG1	DISABLED		
RNG2	DISABLED		
VN	DISABLED		

Network Status: Active

HTML

```
<div style="text-align:center">
 <h1>Fluid Sensor Dashboard by runlinc</h1>
 <br>
 <br>
 Current Status: <font id="Status">Water Level Sensor is loading...</font>
</div>
```

JavaScript Select Macro | select a device | Add Macro

```
var thresholdVoltage = 100;
turnOn( Transmitter );
```

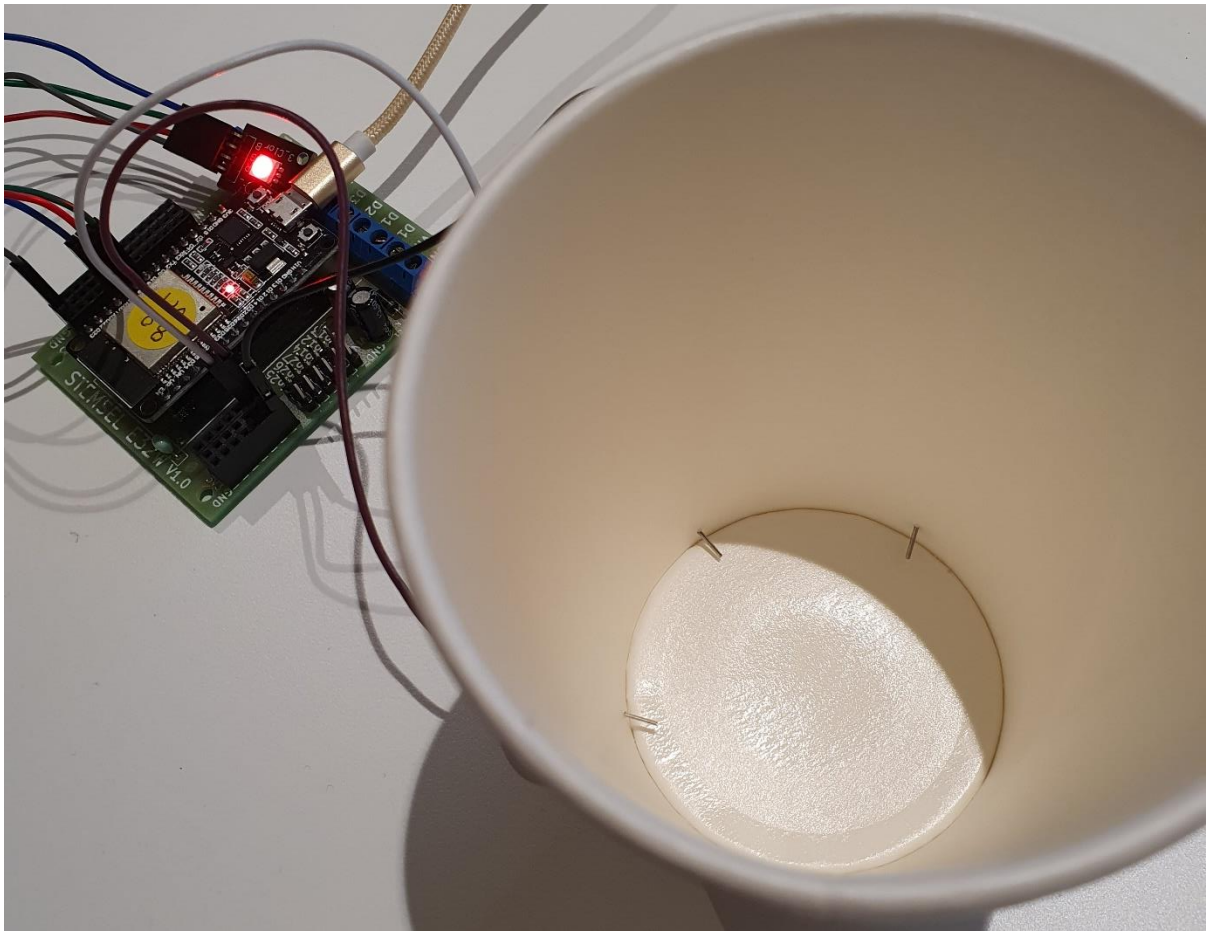
JavaScript Loop Select Macro | select a device | Add Macro

```
receiverValue = analogIn( Receiver );

if(receiverValue >= thresholdVoltage){
turnOn( Green );
turnOff( Red );
document.getElementById("Status").innerHTML = "The water level is above the
marked level.";
}else{
turnOn( Red );
turnOff( Green );
document.getElementById("Status").innerHTML = " The water level is below the
marked level.";
}
await mSec(500);
```

Fluid Sensor Dashboard by runlinc

Current Status: The water level is below the marked level.



Expected Result (Main part): Water in the cup (only filled to overflow the sensor).

TX2	DISABLED		
D18	DIGITAL_OUT	Red	OFF
D19	DIGITAL_OUT	Green	ON
D21	DIGITAL_OUT	Blue	OFF
D22	DISABLED		
D23	DISABLED		
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	ANALOG_IN	Receiver	183
D33	DIGITAL_OUT	Transmitter	ON
D34	DISABLED	UpperTopReceiver	
D35	DISABLED	LowerBottomReceiver	
VP	DISABLED	BottomReceiver	
RNG1	DISABLED		
RNG2	DISABLED		
VN	DISABLED		

HTML

```
<div style="text-align:center">
 <h1>Fluid Sensor Dashboard by runlinc</h1>
 <br>
 <br>
 Current Status: <font id="Status">Water Level Sensor is loading...</font>
</div>
```

JavaScript Select Macro | select a device | Add Macro

```
var thresholdVoltage = 100;
turnOn( Transmitter );
```

JavaScript Loop Select Macro | select a device | Add Macro

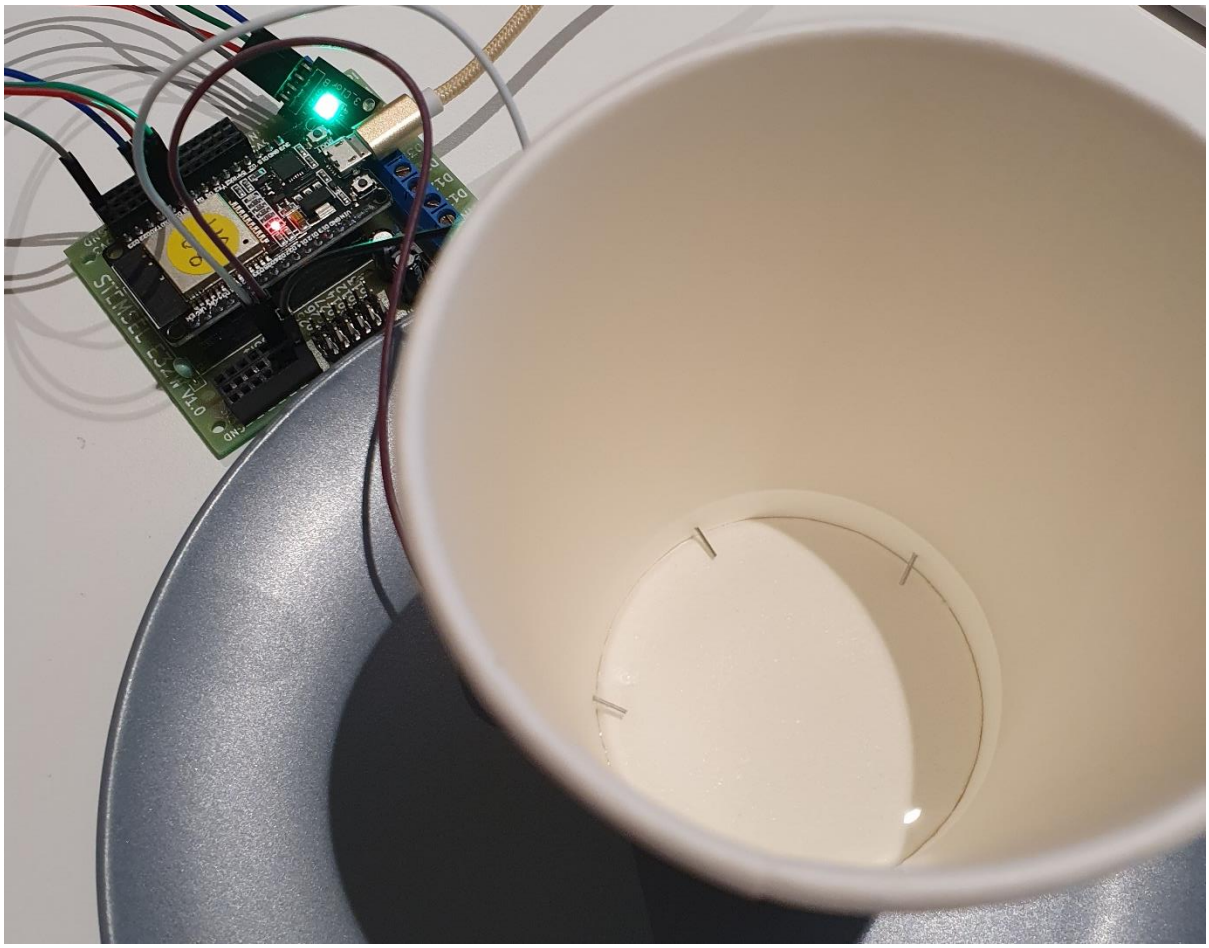
```
receiverValue = analogIn( Receiver );

if(receiverValue >= thresholdVoltage){
turnOn( Green );
turnOff( Red );
document.getElementById("Status").innerHTML = "The water level is above the
marked level.";
}else{
turnOn( Red );
turnOff( Green );
document.getElementById("Status").innerHTML = " The water level is below the
marked level.";
}
await mSec(500);
```

Network Status: Active

Fluid Sensor Dashboard by runlinc

Current Status: The water level is above the marked level.



Part A2: Design the Circuit on runlinc (Extension Setup)

Note: Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Use the left side of the runlinc web page to construct an input/output (I/O).

For extension part, add 3 new extra receivers into the I/O on runlinc, and change the name for port D32.
For port D32 name it TopReceiver and set it as ANALOG_IN.
For port D34 name it UpperTopReceiver and set it as ANALOG_IN.
For port D35 name it LowerBottomReceiver and set it as ANALOG_IN.
For port VP name it BottomReceiver and set it as ANALOG_IN.

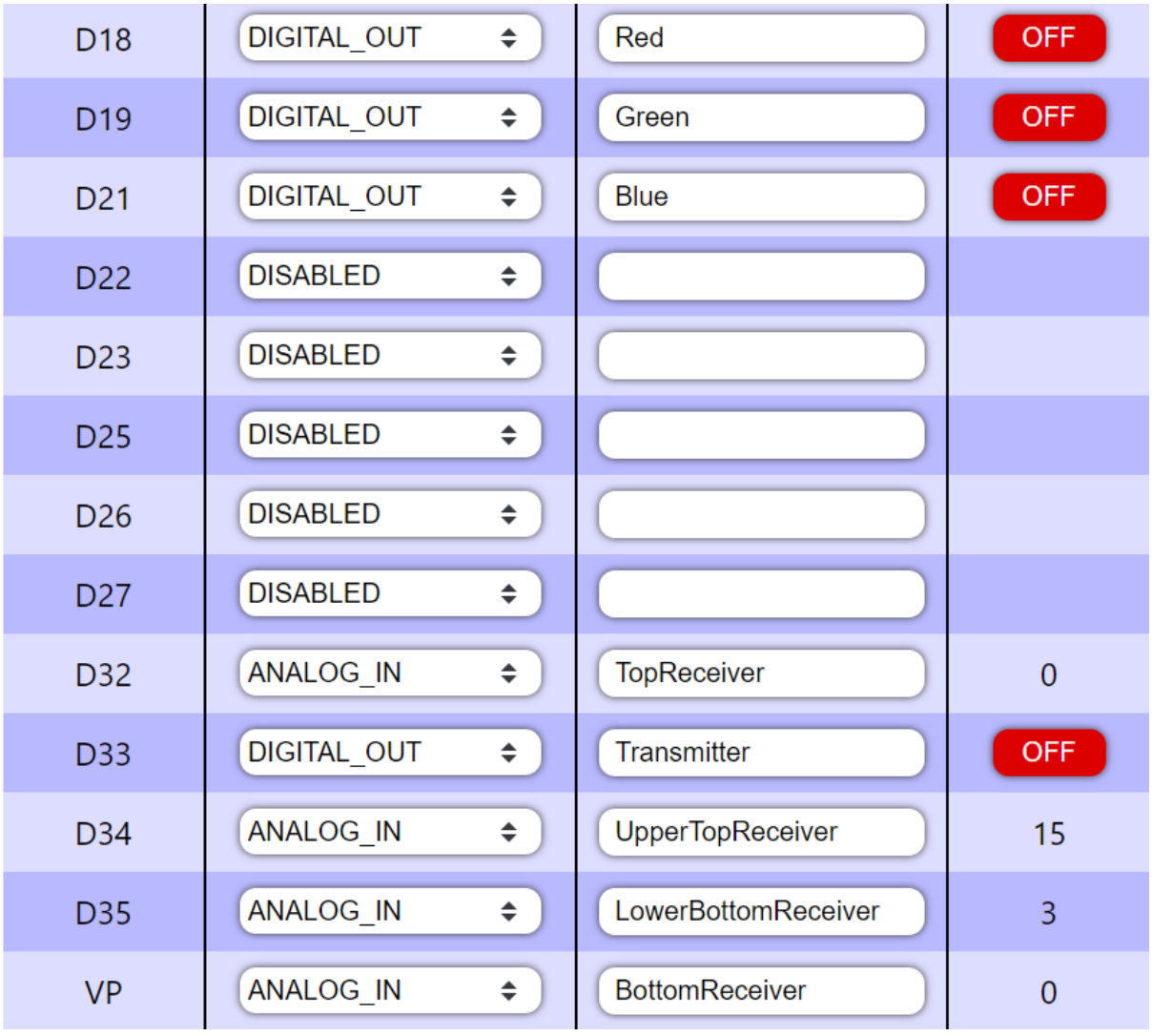


Figure 9: I/O configurations connections

Part B2: Build the Circuit (Extension Setup)

Wiring instructions (Extension Setup)

For the extension part, 3 new jumper wires are added to the board for 3 new receivers. They are added to the signal ports of io34, io35 and VP.

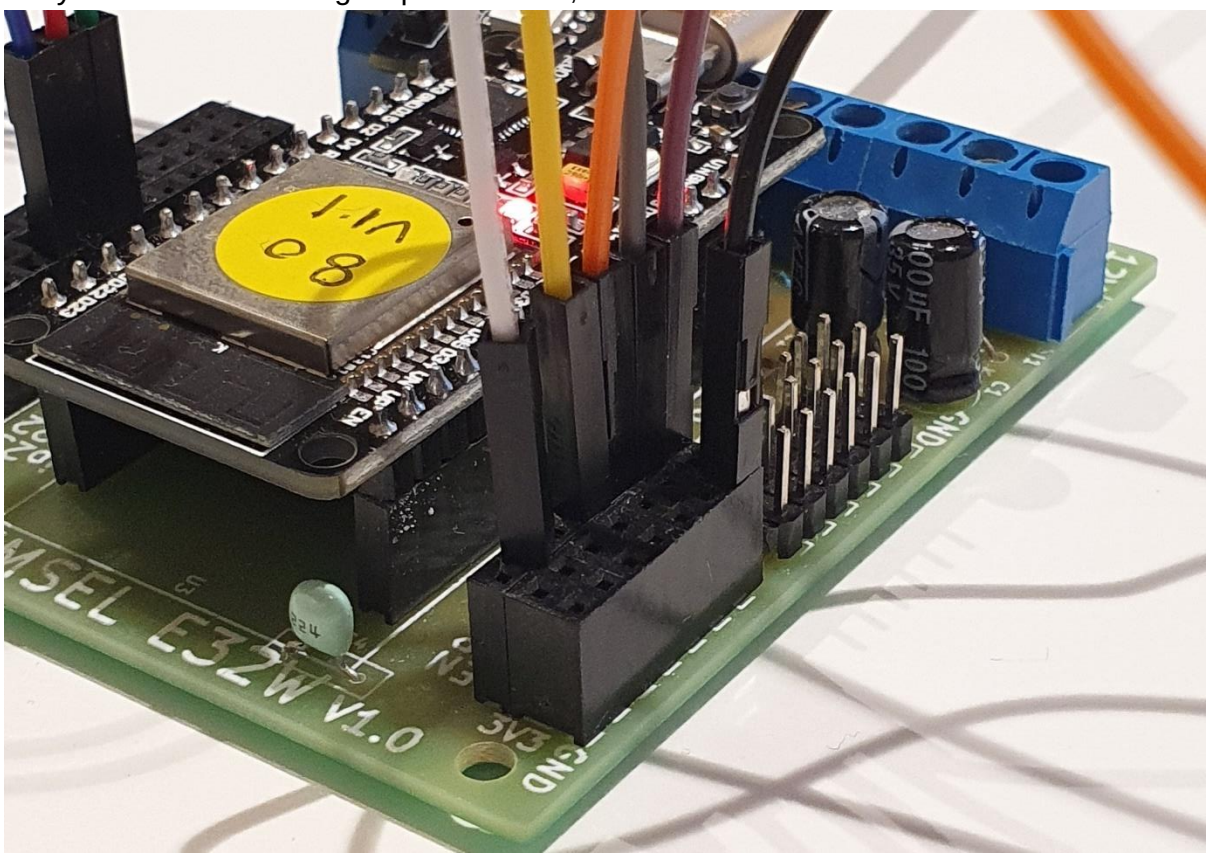


Figure 10: Side View for 3 new jumper wires added to signal ports of io34, io35 and VP.

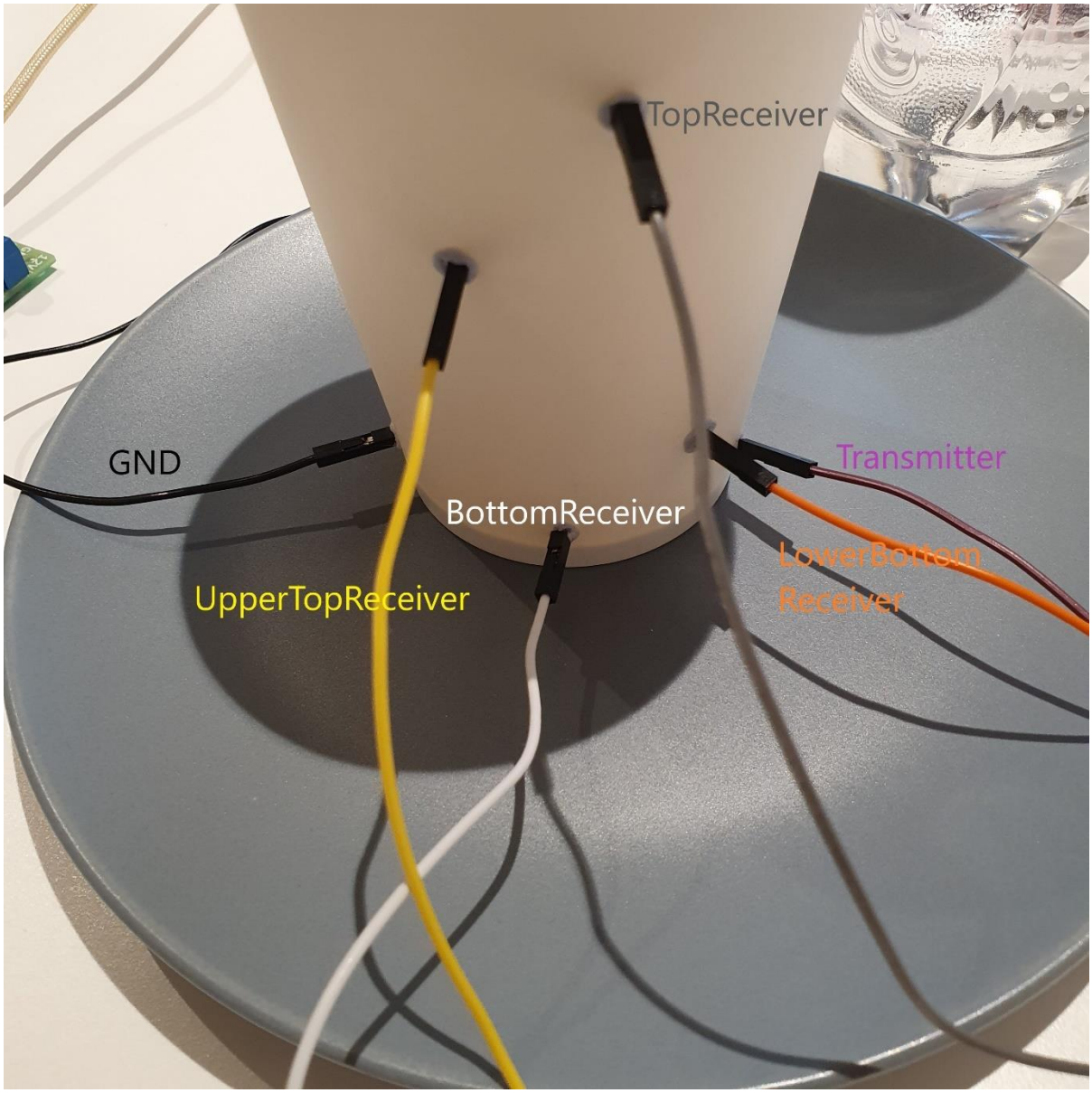


Figure 11: All 6 jumper wires connected to paper cup (labelled).

Part C2: Program the Circuit (Extension Setup)

JavaScript Loop Changes:

1. Since we added 3 new receivers, we need to have a total of 4 new distinct analogIns.

```
B_receiverValue = analogIn( BottomReceiver );
LB_receiverValue = analogIn( LowerBottomReceiver );
LT_receiverValue = analogIn( UpperTopReceiver );
T_receiverValue = analogIn( TopReceiver );
```

2. We will now start checking the fluid level of the cup. First, check if the cup is full. If it is full, turn on colour RED on the KY-009 module and change the status. If the cup is not full, continue to the next IF loop.

```
//Check Threshold from Top to Bottom
//Check if cup is full
if (T_receiverValue >= thresholdVoltage){
turnOn( Red );
document.getElementById("Status").innerHTML = "The cup is CRITICALLY FULL.";
}else{
```

3. We will now start check if the cup is near full. The colour PURPLE (by mixing colour red and blue) will turn on if the cup is near full, and change the status. If not then it will continue to the next IF loop.

```
//Check if cup is near full
if (LT_receiverValue >= thresholdVoltage){
turnOn( Red );
turnOn( Blue );
document.getElementById("Status").innerHTML = "The cup is NEARLY FULL.";
}else{
```

4. We will now start check if the cup is at adequate level. The colour GREEN will turn on if the cup is at an adequate level, and change the status. If not then it will continue to the next IF loop.

```
//Check if cup is at adequate level
if (LB_receiverValue >= thresholdVoltage){
turnOn( Green );
document.getElementById("Status").innerHTML = "The cup is at acceptable level.";
}else{
```

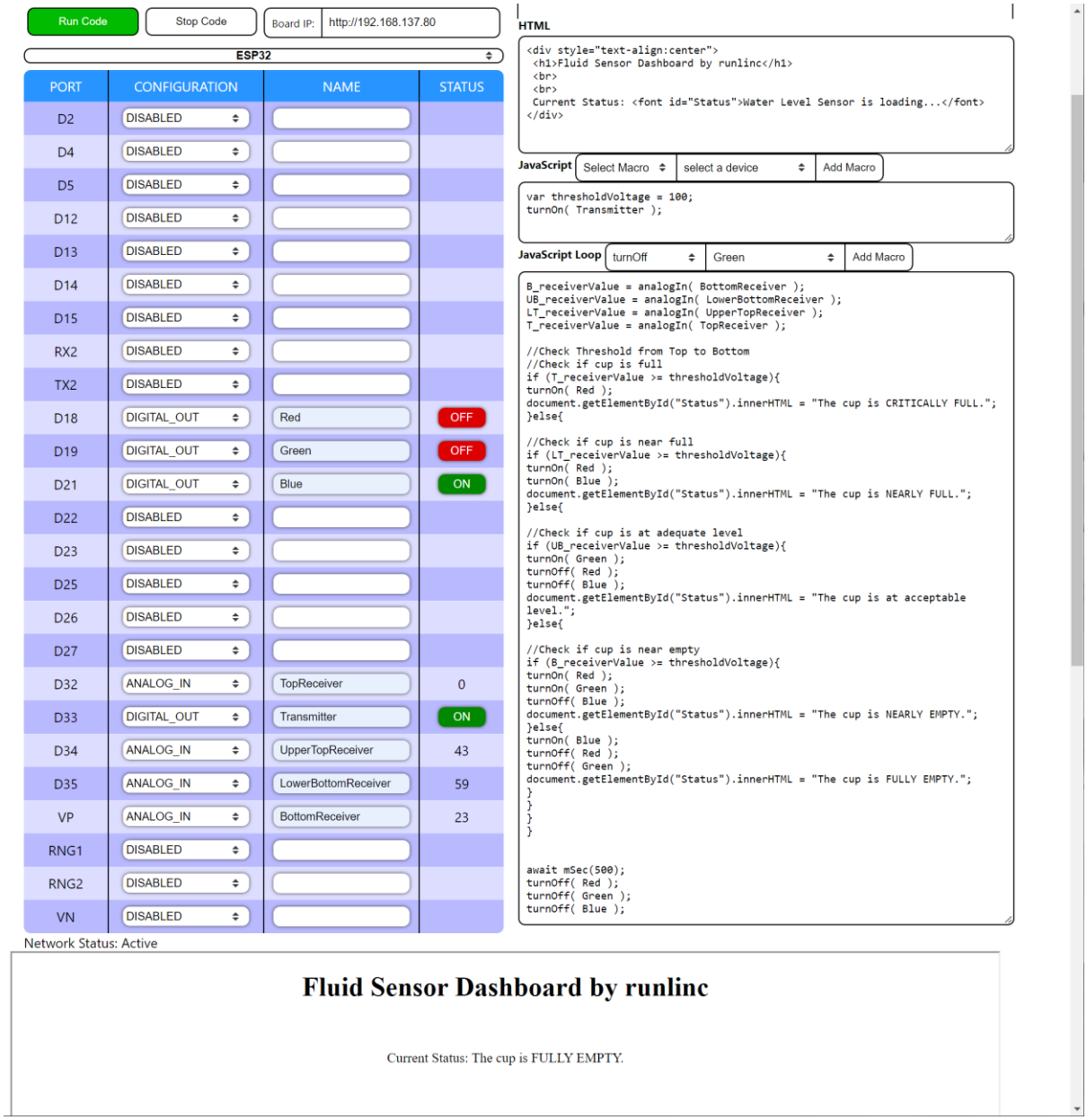
5. We will now start check if the cup is near empty or fully empty. The colour YELLOW (by mixing colour red and green) will turn on if the cup is nearly empty, and change the status. If not then it will turn on the colour BLUE and change the status to fully empty.

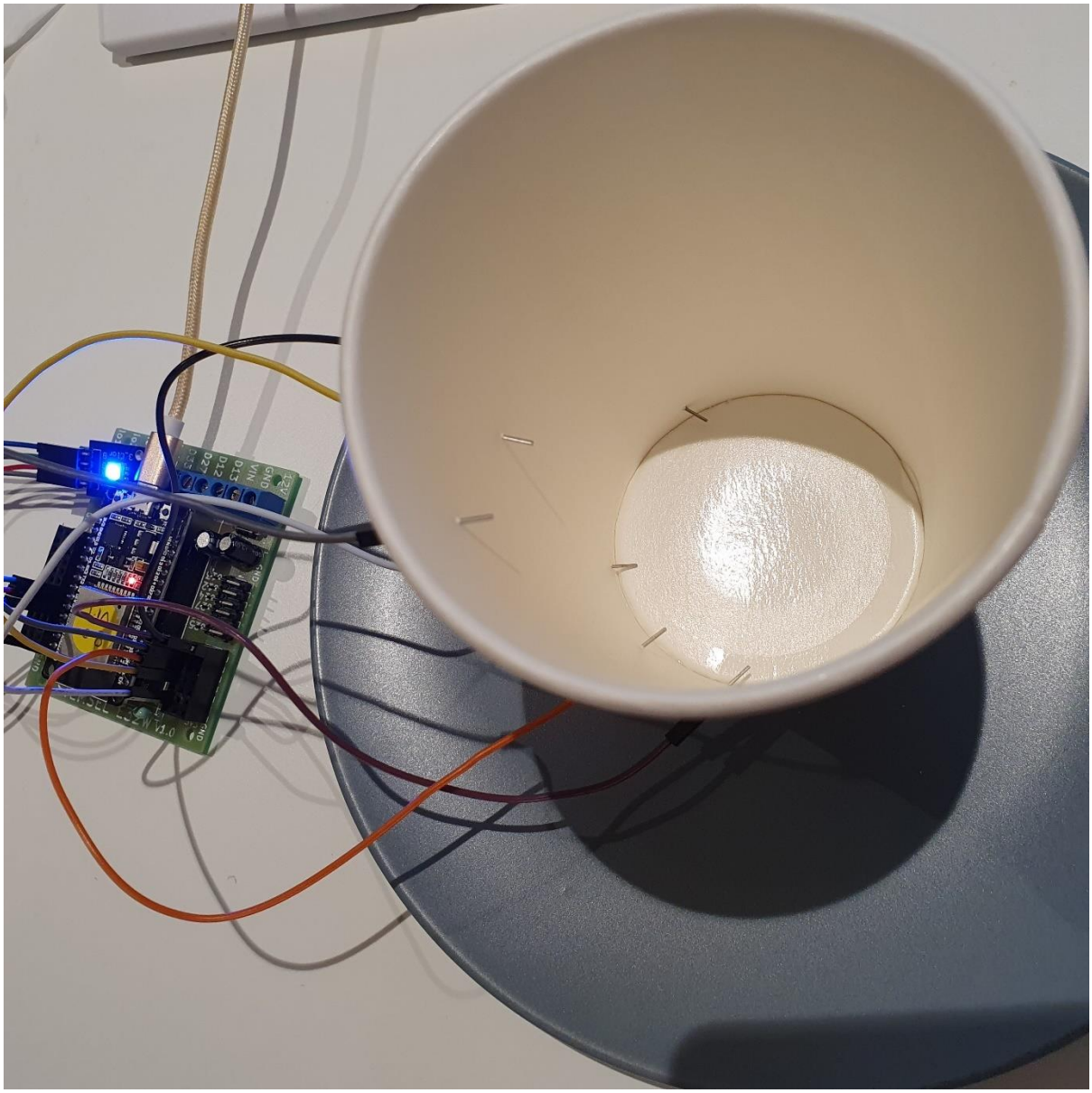
```
//Check if cup is near empty
if (B_receiverValue >= thresholdVoltage){
turnOn( Red );
turnOn( Green );
document.getElementById("Status").innerHTML = "The cup is NEARLY EMPTY.";
}else{
turnOn( Blue );
document.getElementById("Status").innerHTML = "The cup is FULLY EMPTY.";
}
}
}
}
```

6. Now outside of the if-else statement block and back to the main portion of the JavaScript Loop, we will need to add some delay and then turn Off all of the LEDs.

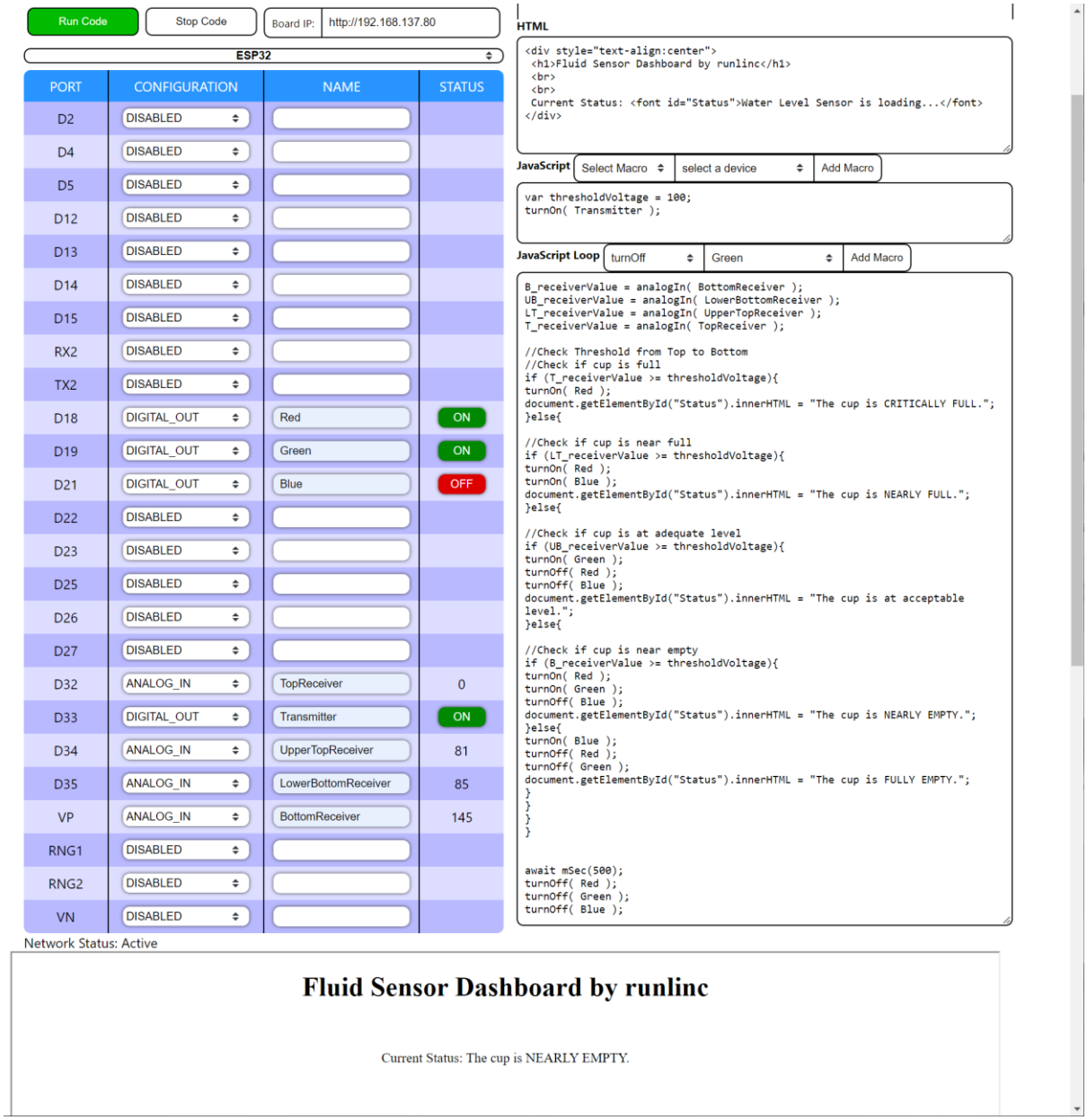
```
await mSec(500);
turnOff( Red );
turnOff( Green );
turnOff( Blue );
```

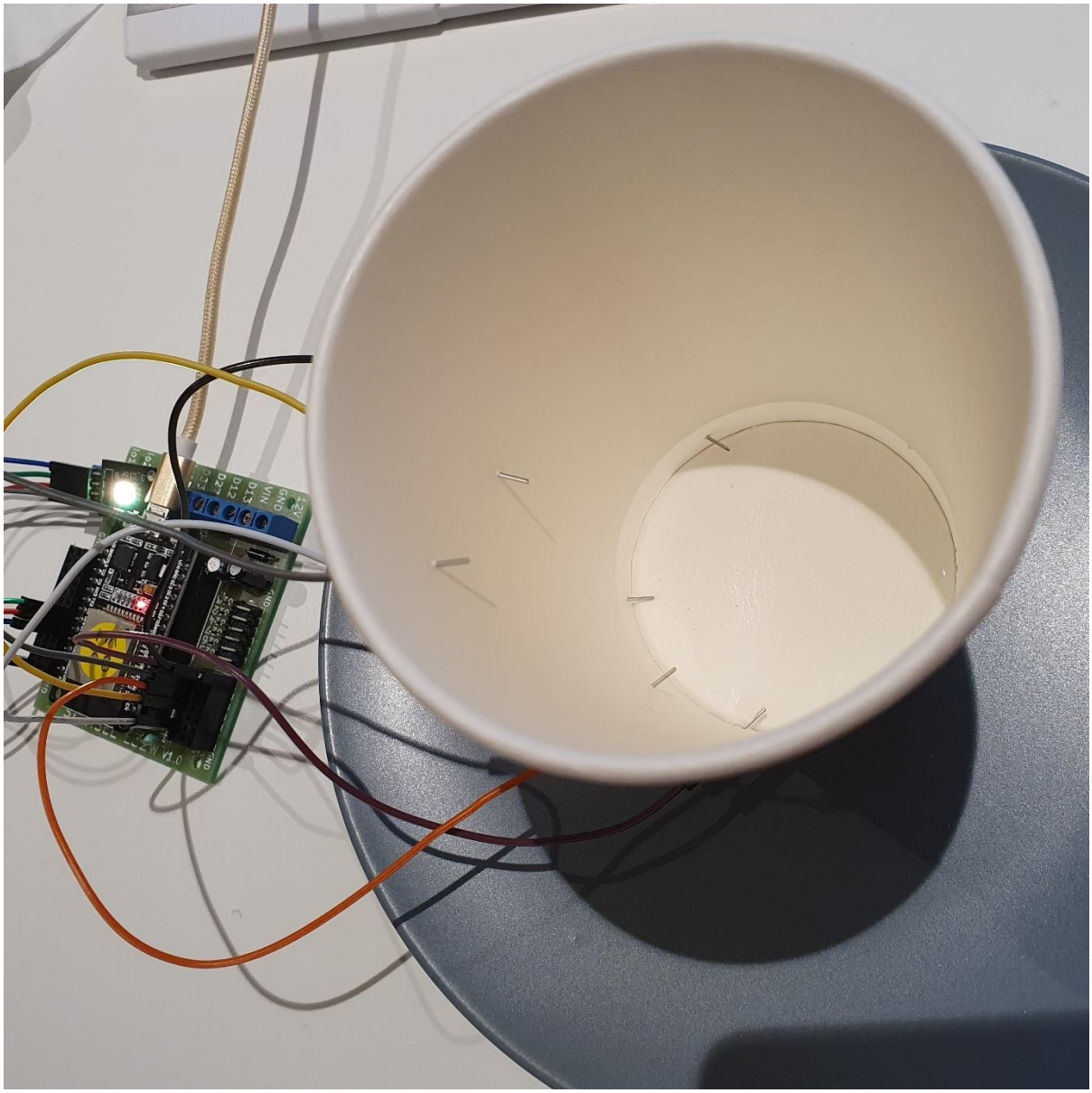
Expected Result (Extension part): No water in the cup.



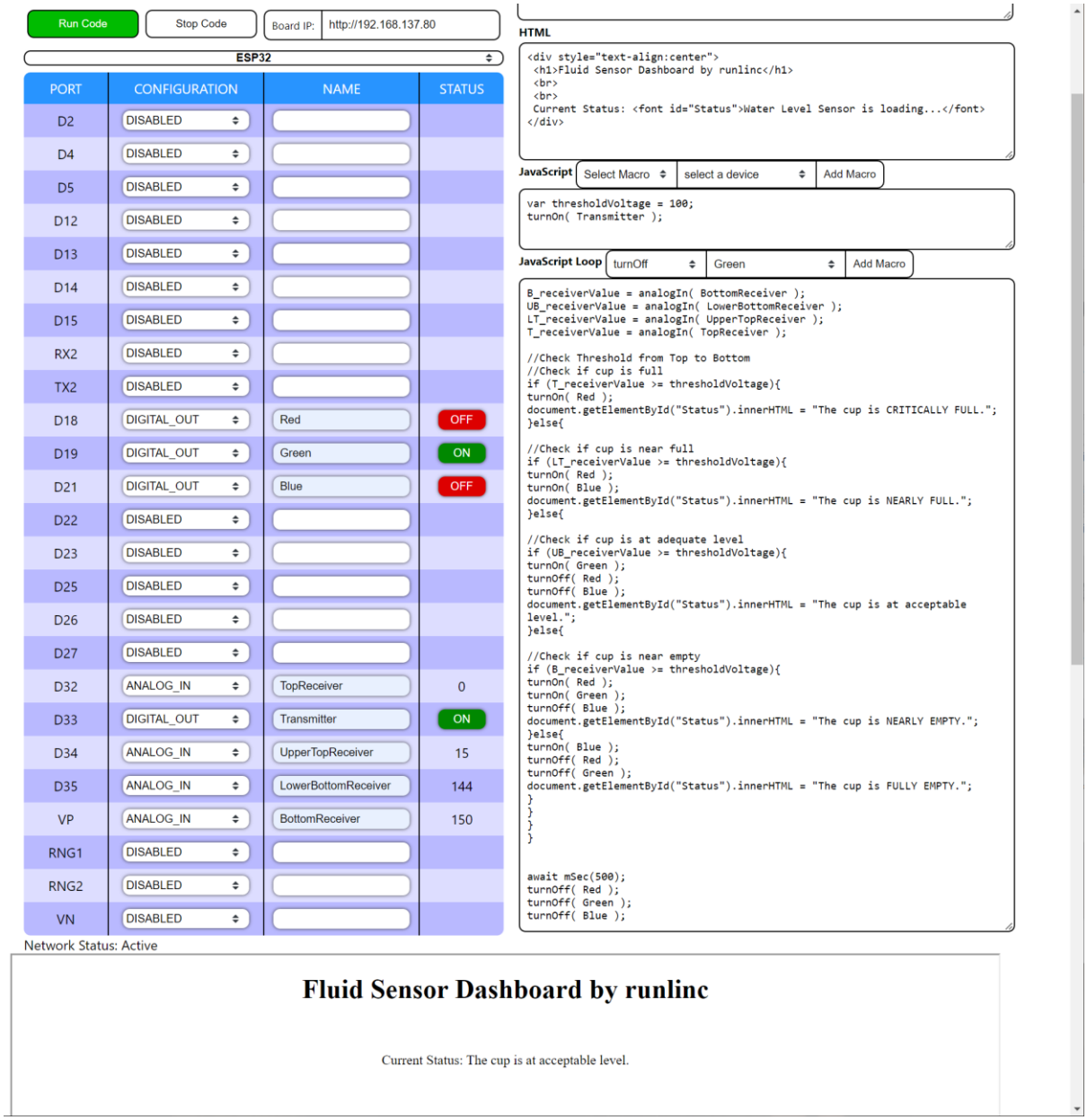


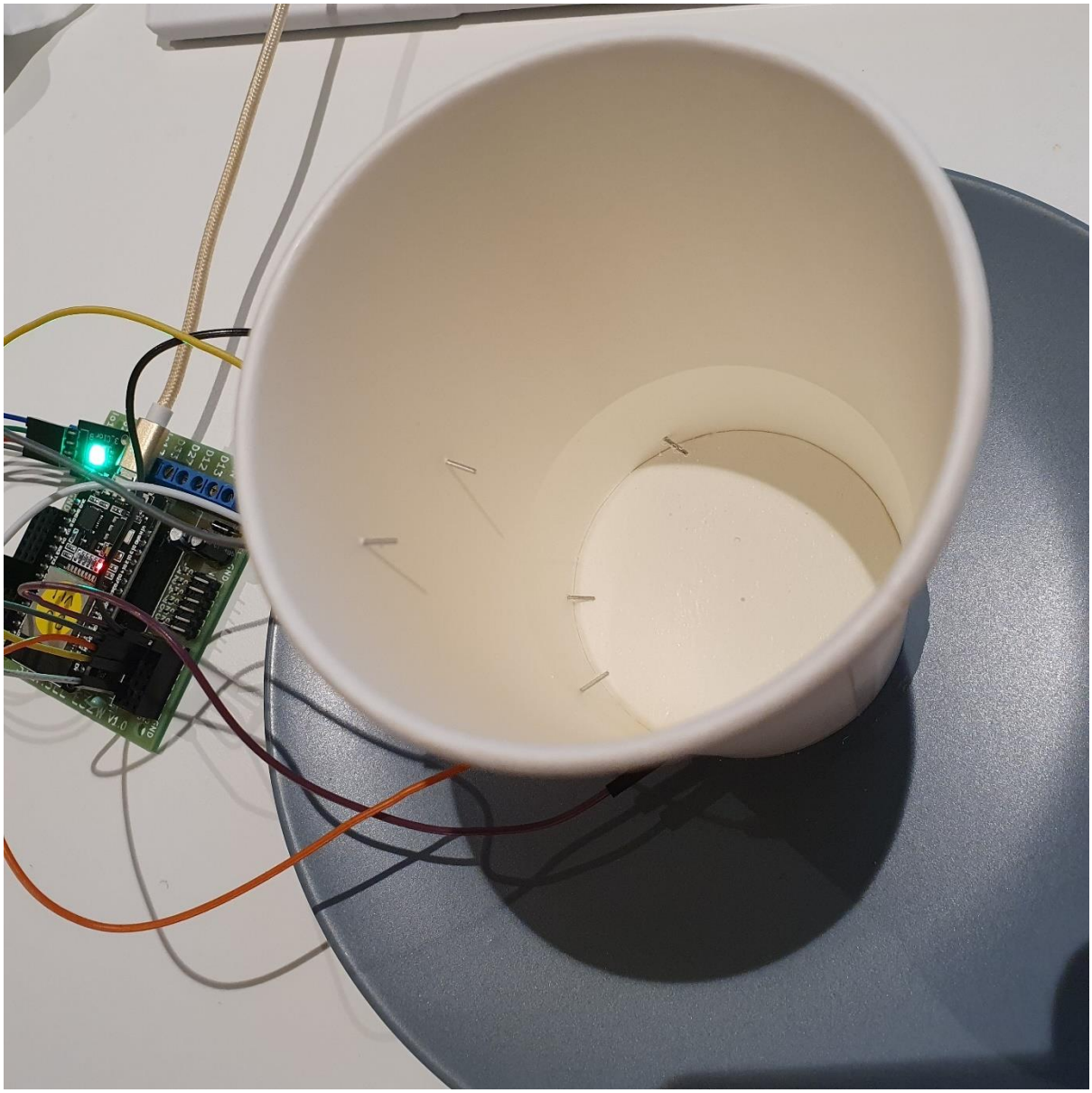
Expected Result (Extension part): Slight water in the cup.



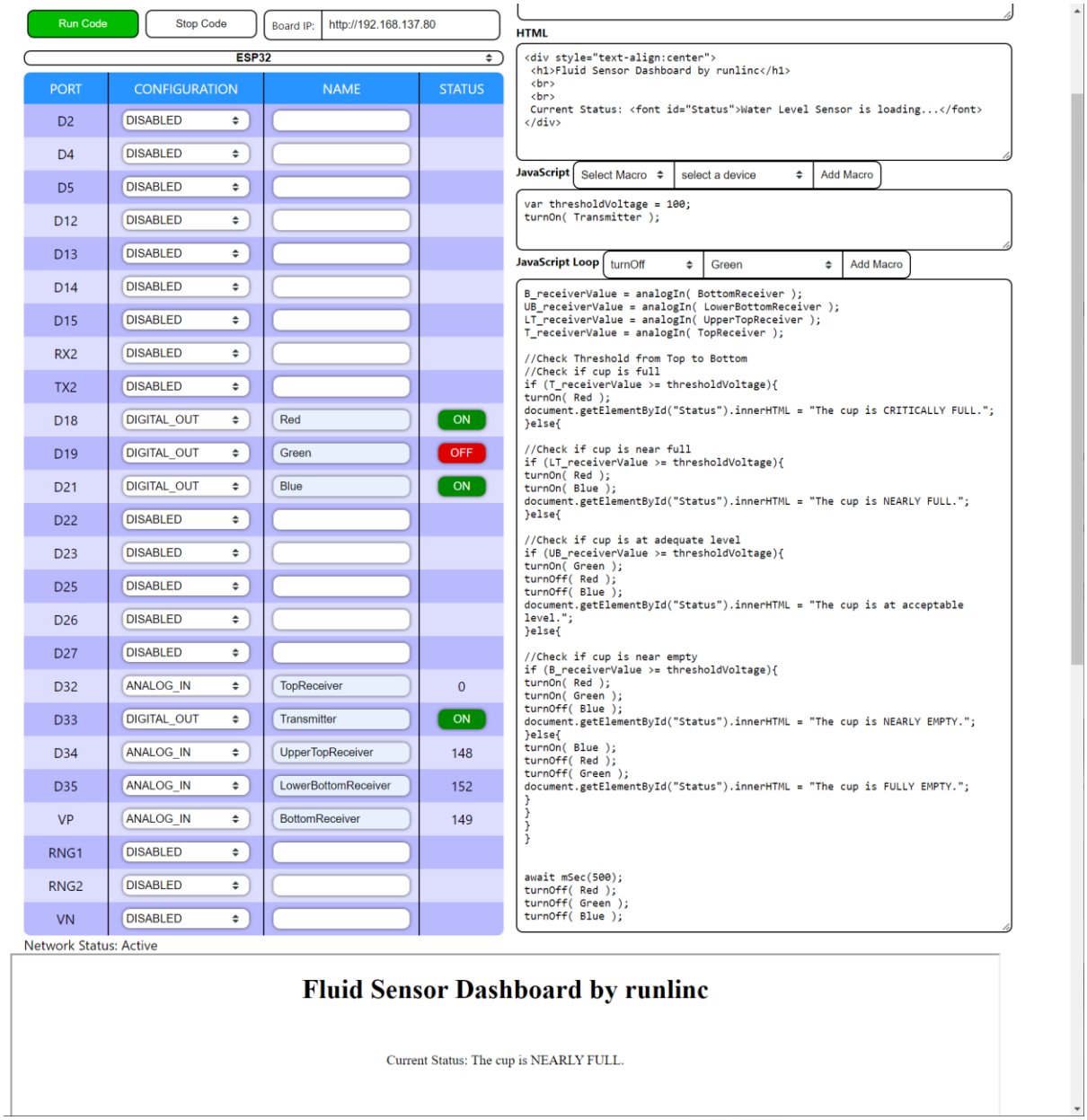


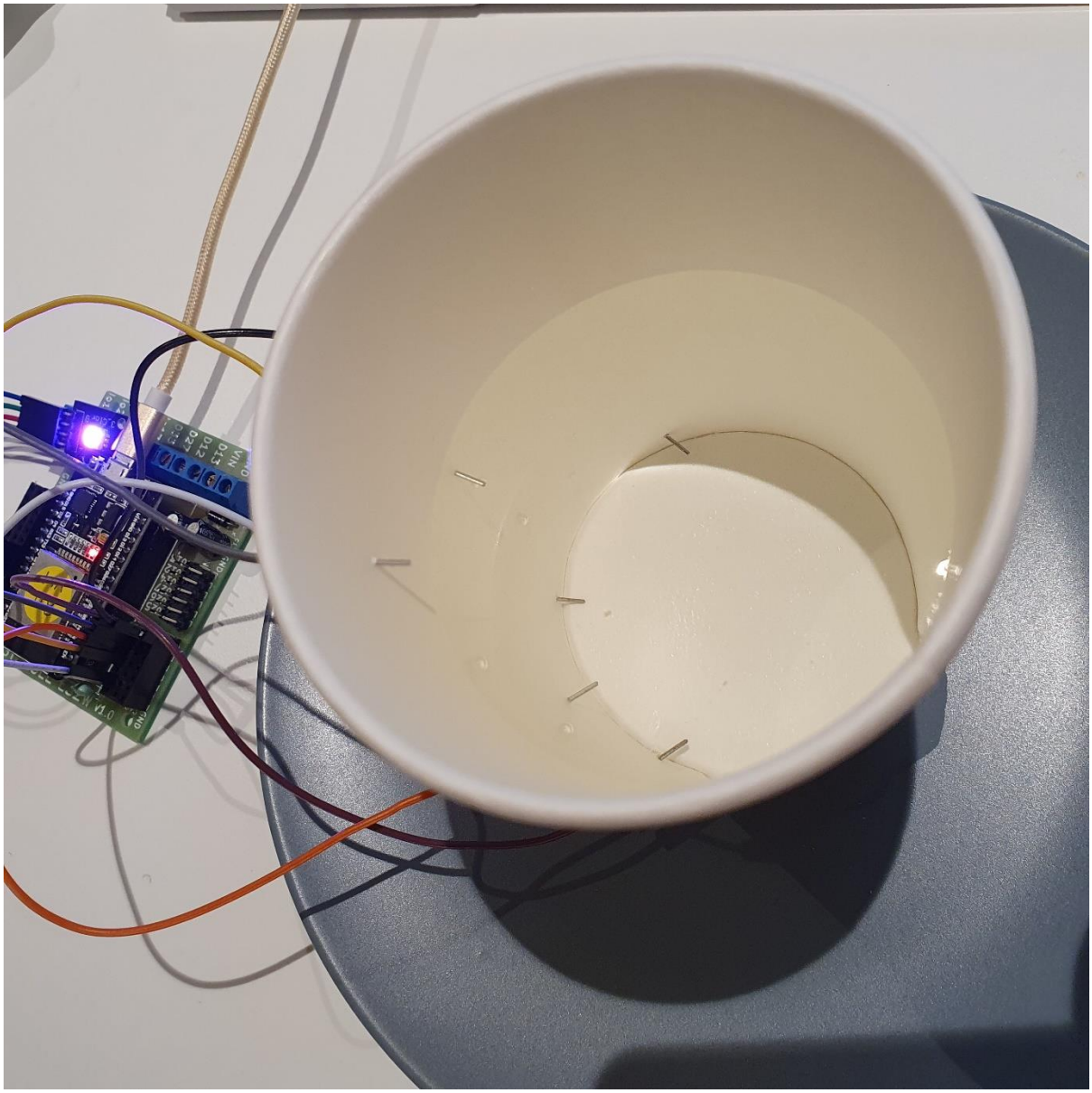
Expected Result (Extension part): Some water in the cup.



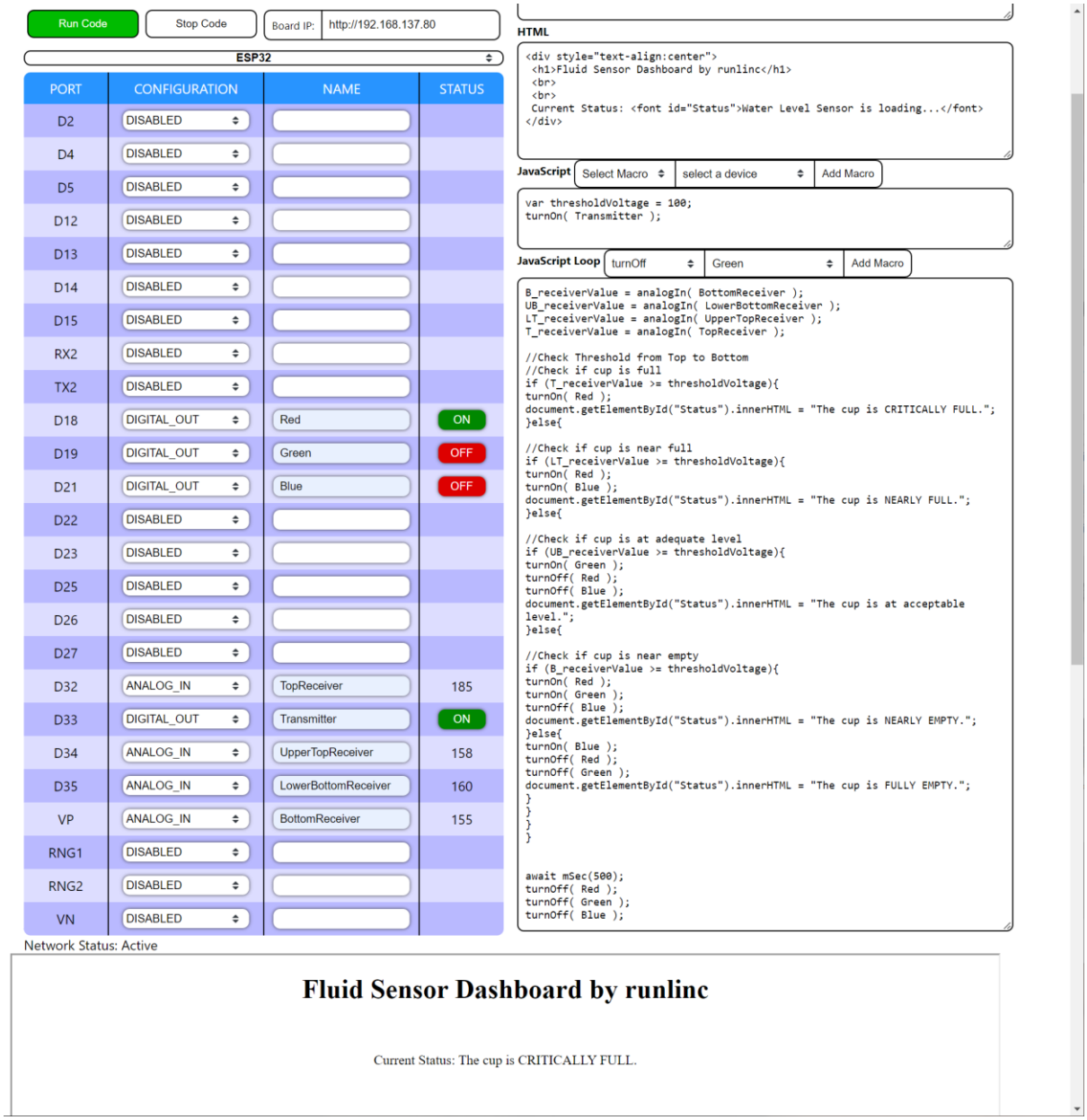


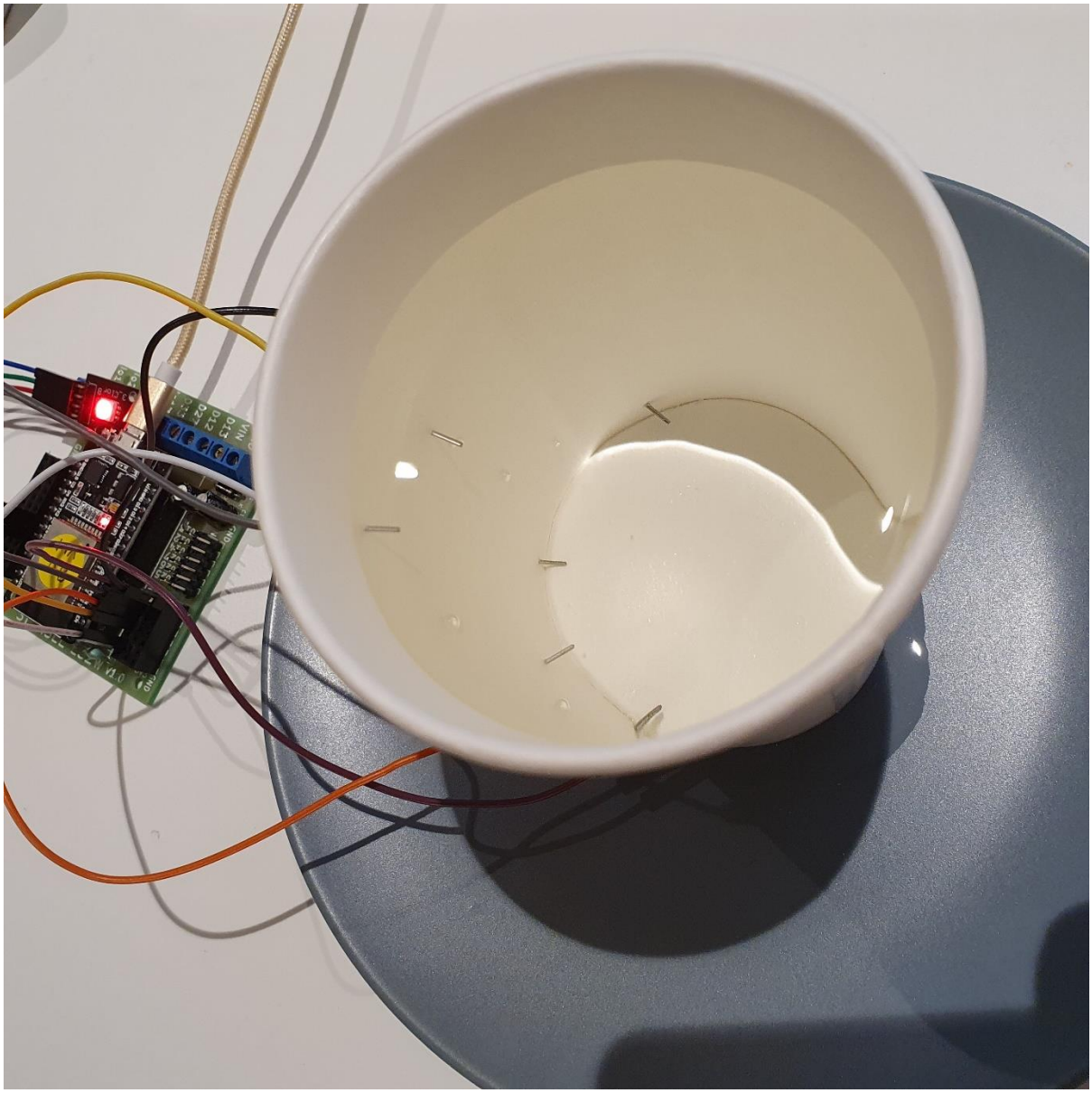
Expected Result (Extension part): Big amount of water in the cup.





Expected Result (Extension part): Huge amount of water in the cup.





Complete Code (main & extension)

HTML:

```
<div style="text-align:center">
 <h1>Fluid Sensor Dashboard by runlinc</h1>
 <br>
 <br>
 Current Status: <font id="Status">Water Level Sensor is loading...</font>
</div>
```

JAVASCRIPT:

```
var thresholdVoltage = 100;
turnOn( Transmitter );
```

JAVASCRIPT LOOP (Main):

```
receiverValue = analogIn( Receiver );

if(receiverValue >= thresholdVoltage){
turnOn( Green );
turnOff( Red );
document.getElementById("Status").innerHTML = "The water level is above the marked level.";
}else{
turnOn( Red );
turnOff( Green );
document.getElementById("Status").innerHTML = " The water level is below the marked level.";
}
await mSec(500);
```

JAVASCRIPT LOOP (Extension):

```
B_receiverValue = analogIn( BottomReceiver );
LB_receiverValue = analogIn( LowerBottomReceiver );
LT_receiverValue = analogIn( UpperTopReceiver );
T_receiverValue = analogIn( TopReceiver );

//Check Threshold from Top to Bottom
//Check if cup is full
if (T_receiverValue >= thresholdVoltage){
turnOn( Red );
document.getElementById("Status").innerHTML = "The cup is CRITICALLY FULL.";
}else{

//Check if cup is near full
if (LT_receiverValue >= thresholdVoltage){
turnOn( Red );
turnOn( Blue );
document.getElementById("Status").innerHTML = "The cup is NEARLY FULL.";
}else{

//Check if cup is at adequate level
if (LB_receiverValue >= thresholdVoltage){
turnOn( Green );
document.getElementById("Status").innerHTML = "The cup is at acceptable level.";
}else{

//Check if cup is near empty
if (B_receiverValue >= thresholdVoltage){
turnOn( Red );
turnOn( Green );
document.getElementById("Status").innerHTML = "The cup is NEARLY EMPTY.";
}else{
turnOn( Blue );
document.getElementById("Status").innerHTML = "The cup is FULLY EMPTY.";
}
}
}
}


await mSec(500);
turnOff( Red );
turnOff( Green );
turnOff( Blue );
```

Challenge

Although warning lights are good to notify someone if their tank is running low, they might not always see them with all the other lights and notifications on a dashboard. What other notifications are there? Add either code or hardware that would give the farmer another form of feedback besides visual to warn them their tank level.

Summary

By using some wires, LED's and a microchip, we were able to measure the level of fluid in a cup. During this project, we learned how can we use the microchip to compare the voltage of wires in a liquid. Although this project only used a small cup, the principles can be applied to a fluid tank. This is a connection of STEMSEL. A small project that teaches how to turn some LED's on and off and program a microchip can be applied to real-world applications to improve the industry and improve the lives of people around the world.